# On Multiset Selection with Size Constraints*

**Chao Qian**[1], **Yibo Zhang**[1], **Ke Tang**[2], **Xin Yao**[2]
[1]Anhui Province Key Lab of Big Data Analysis and Application, School of Computer Science and Technology,
University of Science and Technology of China, Hefei 230027, China
[2]Shenzhen Key Lab of Computational Intelligence, Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China
chaoqian@ustc.edu.cn, zyb233@mail.ustc.edu.cn, {tangk3, xiny}@sustc.edu.cn

## Abstract

This paper considers the multiset selection problem with size constraints, which arises in many real-world applications such as budget allocation. Previous studies required the objective function $f$ to be submodular, while we relax this assumption by introducing the notion of the submodularity ratios (denoted by $\alpha_f$ and $\beta_f$). We propose an anytime randomized iterative approach POMS, which maximizes the given objective $f$ and minimizes the multiset size simultaneously. We prove that POMS using a reasonable time achieves an approximation guarantee of $\max\{1-e^{-\beta_f}, (\alpha_f/2)(1-e^{-\alpha_f})\}$. Particularly, when $f$ is submdoular, this bound is at least as good as that of the previous greedy-style algorithms. In addition, we give lower bounds on the submodularity ratio for the objectives of budget allocation. Experimental results on budget allocation as well as a more complex application, namely, generalized influence maximization, exhibit the superior performance of the proposed approach.

## Introduction

The subset selection problem, which selects a subset of size at most $k$ from a total set of $n$ items for maximizing some given objective function $f$, arises in many applications, e.g., maximum coverage (Feige 1998), sensor placement (Krause, Singh, and Guestrin 2008) and influence maximization (Kempe, Kleinberg, and Tardos 2003). It is generally NP-hard. When the objective function $f$ satisfies the monotone and submodular property, the greedy algorithm can achieve a tight approximation guarantee of $1-1/e$ (Nemhauser, Wolsey, and Fisher 1978).

However, in many practical applications such as budget allocation (Alon, Gamzu, and Tennenholtz 2012) and welfare maximization (Kapralov, Post, and Vondrák 2013), a generalization of subset selection, i.e., multiset selection, has to be considered. That is, an item can be selected by multiple times instead of only once, and the objective function is defined over a multiset instead of a subset. Moreover, many subset selection problems can be naturally generalized to this framework. For example, for sensor placement, the number of times of selecting one sensor can correspond to the energy level of the sensor (Soma and Yoshida 2015).

Previous studies on multiset selection assumed that the objective function is submodular. Note that for functions defined over a multiset, submodularity does not imply the diminishing return property, called DR-submodularity (Soma and Yoshida 2016). The latter is stronger, although they are equivalent for functions over a subset. Soma et al. (2014) first proved that for monotone DR-submodular objective functions, the greedy algorithm, which iteratively selects one item with the largest marginal gain, can achieve a $(1-1/e)$-approximation guarantee. When the objective function is relaxed to be monotone submodular, Alon et al. (2012) showed that the generalized greedy algorithm, which can select multiple copies of the same item simultaneously in one iteration, achieves a $(1/2)(1-1/e)$-approximation guarantee. This can be further improved to $(1-1/e)$ by partial enumerations, but with an impractical computation time (Soma et al. 2014). Note that the runtime of these algorithms is polynomial w.r.t. the budget $k$. In (Soma and Yoshida 2016), algorithms with runtime polynomial w.r.t. $\log k$ were developed, but with a loss $\epsilon > 0$ on the approximation ratio.

Besides size constraints, more complex constraints (e.g., knapsack and polymatroid) were also studied in (Soma et al. 2014; Soma and Yoshida 2016). Recently, multiset selection without constraints has been considered. Gottschalk and Peis (2015) proposed a $(1/3)$-approximation algorithm for maximizing submodular functions; while for maximizing DR-submodular functions, a $(1/(2+\epsilon))$-approximation algorithm was provided in (Soma and Yoshida 2017).

This paper relaxes the assumption on submodularity. We study the problem of Multiset Selection with size constraints, where the objective function is only required to be monotone, not necessarily submodular. We propose a Pareto Optimization method for this problem, briefly called POMS. The POMS method first reformulates the original problem as a bi-objective optimization problem that maximizes the given objective $f$ and minimizes the multiset size simultaneously, then employs a randomized iterative algorithm to solve it, and finally selects the best solution satisfying the size constraint from the produced set of solutions.

We theoretically investigate the performance of POMS.

*This work was supported by the National Key Research and Development Program of China (2017YFB1003100, 2017YFB1003102), the NSFC (61603367, 61672478), the YESS (2016QNRC001) and the Science and Technology Innovation Committee Foundation of Shenzhen (ZDSYS201703031748284).

The main contributions can be summarized as follows:

- We introduce the notion of the submodularity ratio $\alpha_f \in [0,1]$ and the DR-submodularity ratio $\beta_f \in [0,1]$ (i.e., **Definitions 7 and 8**), which characterize how close a general function $f$ is to submodularity and DR-submodularity, respectively. Note that $\alpha_f \geq \beta_f$. To introduce $\alpha_f$, we build an equivalent relation between submodularity and a weak version of DR-submodularity (i.e., **Lemma 2**).
- We prove that POMS using a reasonable time can achieve an approximation guarantee of $\max\{1 - e^{-\beta_f}, (\alpha_f/2)(1-e^{-\alpha_f})\}$ (i.e., **Theorem 1**).
- When $f$ is DR-submodular (where $\alpha_f = \beta_f = 1$), the approximation bound of POMS becomes $1-1/e$, which is as same as the best known one, previously obtained by the greedy algorithm (Soma et al. 2014). When $f$ is submodular (where $\alpha_f = 1$), the bound becomes $\max\{1-e^{-\beta_f}, (1-1/e)/2\}$, which is at least as good as that of the generalized greedy algorithm (i.e., $(1-1/e)/2$ (Alon, Gamzu, and Tennenholtz 2012)). For the submodular real-world application, budget allocation, we further provide lower bounds on the DR-submodularity ratio $\beta_f$ of the corresponding objective functions (i.e., **Lemmas 5 and 6**).

We also empirically investigate the actual performance of POMS. We compare it with the greedy-style algorithms on budget allocation and a more complex application, generalized influence maximization. The experimental results on real-world data sets show the better performance of POMS.

The rest of the paper first introduces the studied problem, and then presents the proposed method, its theoretical analysis and empirical study. Finally we conclude this paper.

## Preliminaries

**The General Problem.** Let $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote the set of reals, non-negative reals and non-negative integers, respectively. Given a finite set $V = \{v_1, v_2, \dots, v_n\}$, we study the functions $f : \mathbb{Z}_+^V \to \mathbb{R}$ over the integer lattice $\mathbb{Z}_+^V$, or equivalently over multisets of $V$. Note that a multiset can be naturally represented by a vector $\boldsymbol{x} \in \mathbb{Z}_+^V$, where for the $i$-th entry $x_i$ of $\boldsymbol{x}$, $x_i > 0$ means that the item $v_i$ appears $x_i$ times in the multiset, and $x_i = 0$ means that $v_i$ does not appear.

We give some notations. Let $\boldsymbol{x} \wedge \boldsymbol{y}$ and $\boldsymbol{x} \vee \boldsymbol{y}$ denote the coordinate-wise minimum and maximum, respectively, that is, $\boldsymbol{x} \wedge \boldsymbol{y} = (\min\{x_1, y_1\}, \dots, \min\{x_n, y_n\})$ and $\boldsymbol{x} \vee \boldsymbol{y} = (\max\{x_1, y_1\}, \dots, \max\{x_n, y_n\})$. We denote the size of a multiset by $|\boldsymbol{x}| = \sum_{i=1}^n x_i$. The $i$-th unit vector is denoted by $\boldsymbol{\chi_i}$, that is, the $i$-th entry of $\boldsymbol{\chi_i}$ is 1 and others are 0; the all-zeros and all-ones vectors are denoted by $\mathbf{0}$ (i.e., $\emptyset$) and $\mathbf{1}$, respectively. Let $[n]$ denote the set $\{1, 2, \dots, n\}$.

A function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is monotone if for any $\boldsymbol{x} \leq \boldsymbol{y}$, $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$. Without loss of generality, we assume that monotone functions are normalized, i.e., $f(\mathbf{0}) = 0$. For a function $f : \mathbb{Z}_+^V \to \mathbb{R}$, submodularity (as presented in Definition 1) does not imply the diminishing return property (called DR-submodularity as presented in Definition 2). DR-submodularity is stronger than submodularity, that is, a DR-submodular function is submodular, but not vice versa. Note that for a set function $f : 2^V \to \mathbb{R}$, they are equivalent.

**Definition 1** (Submodular (Soma et al. 2014))**.** *A function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is submodular if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}_+^V$,*

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq f(\boldsymbol{x} \wedge \boldsymbol{y}) + f(\boldsymbol{x} \vee \boldsymbol{y}).$$

**Definition 2** (DR-Submodular (Soma and Yoshida 2016))**.** *A function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is DR-submodular if for any $\boldsymbol{x} \leq \boldsymbol{y}$ and $i \in [n]$,*

$$f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x}) \geq f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y}).$$

Our studied problem as presented in Definition 3 is to maximize a monotone function $f$ with an upper limit on $|\boldsymbol{x}|$. Note that $\boldsymbol{c}$ limits the maximum value on each entry of $\boldsymbol{x}$. If $\boldsymbol{c} = \mathbf{1}$, $\boldsymbol{x} \in \{0,1\}^n$ actually represents a subset of $V$. When the objective function $f$ is DR-submodular, Soma et al. (2014) proved that the greedy algorithm can obtain a $(1-1/e)$-approximation guarantee. The greedy algorithm iteratively selects one item with the largest improvement on $f$. When $f$ is submodular, the generalized greedy algorithm was shown able to achieve a $(1-1/e)/2$-approximation guarantee (Alon, Gamzu, and Tennenholtz 2012). In each iteration, it selects a combination $(v_i, j)$ such that the average marginal gain by adding $j$ copies of $v_i$ (i.e., $x_i = x_i + j$) is maximized; the better of the found multiset $\boldsymbol{x}$ and the best $j\boldsymbol{\chi_i}$ is finally returned. Note that by partial enumerations, the generalized greedy algorithm can even achieve a $(1-1/e)$-approximation guarantee, but the runtime is impractical.

**Definition 3** (The General Problem)**.** *Given a monotone objective function $f : \mathbb{Z}_+^V \to \mathbb{R}_+$, a vector $\boldsymbol{c} \in \mathbb{Z}_+^V$ and a budget $k \in \mathbb{Z}_+$, it is to find a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$ such that*

$$\arg\max\nolimits_{\boldsymbol{x} \leq \boldsymbol{c}} \quad f(\boldsymbol{x}) \quad s.t. \quad |\boldsymbol{x}| \leq k. \tag{1}$$

Here are some examples, that will be investigated.

**Budget Allocation.** Let a bipartite graph $G = (V, T; E)$ represent a social network, where each source node in $V$ is a marketing channel, each target node in $T$ is a customer, and $E \subseteq V \times T$ is the edge set. The goal of budget allocation is to distribute the budget $k$ among the source nodes such that the expected number of target nodes that get activated is maximized (Alon, Gamzu, and Tennenholtz 2012). The allocation of the budget $k$ can be represented by a vector $\boldsymbol{x} \in \mathbb{Z}_+^V$, where $x_i$ is the budget allocated on $v_i \in V$. Each source node $v_i$ ($i \in [n]$) has a capacity $c_i$ (i.e., $x_i \leq c_i$) and probabilities $p_i^{(j)} \in [0,1]$ for $j \in \{1, 2, \dots, c_i\}$. If $x_i > 0$, the source node $v_i$ will make $x_i$ independent trials to activate each neighboring target node $t$, where the probability of activating $t$ in the $j$-th trial is $p_i^{(j)}$. Thus, the probability that a target node $t \in T$ gets activated is

$$f_t(\boldsymbol{x}) = 1 - \prod_{i:(v_i,t) \in E} \prod_{j=1}^{x_i} (1 - p_i^{(j)}). \tag{2}$$

By the linearity of expectation, the expected number of active target nodes is $\sum_{t \in T} f_t(\boldsymbol{x})$, which is monotone and submodular (Alon, Gamzu, and Tennenholtz 2012). When $p_i^{(j)}$ is nonincreasing with $j$ (i.e., $p_i^{(j)} \geq p_i^{(j+1)}$), the objective $\sum_{t \in T} f_t(\boldsymbol{x})$ is DR-submodular (Soma et al. 2014).

**Definition 4** (Budget Allocation)**.** *Given a bipartite graph $G = (V, T; E)$, capacities $c_i$ $(i \in [n])$, probabilities $p_i^{(j)}$ $(i \in [n],\ j \in \{1, 2, \dots, c_i\})$, and a budget $k$, it is to find a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$ such that*

$$\arg\max\nolimits_{\boldsymbol{x} \le \boldsymbol{c}} \quad \sum\nolimits_{t \in T} f_t(\boldsymbol{x}) \quad s.t. \quad |\boldsymbol{x}| \le k,$$

*where $f_t(\boldsymbol{x})$ is defined as Eq. (2).*

**Budget Allocation with a Competitor.** In (Soma et al. 2014), budget allocation is extended to the two-player case: a competitor against an advertiser. The budget of the competitor is allocated in advance, and the advertiser aims at allocating the budget such that the expected number of target nodes activated by its trials is maximized. The competitor and the advertiser propagate in a discrete time step; in each trial of propagation, the competitor performs before the advertiser. Let $\boldsymbol{x} \in \mathbb{Z}_+^V$ denote the budget allocation of the advertiser. For the advertiser in the $j$-th trial, each source node $v_i$ with $x_i \ge j$ will activate each neighboring target node $t$ with probability $p_i^{(j)}$ if $t$ is inactive and with probability $q_i^{(j)}$ $(q_i^{(j)} \le p_i^{(j)})$ if $t$ has been activated by the competitor. Let $E_{t,l}$ denote the event that the target node $t$ is activated by the competitor in the $l$-th trial. Thus, conditioned on $E_{t,l}$, the probability that a target node $t \in T$ gets activated by the advertiser is

$$f_{t,l}(\boldsymbol{x}) = 1 - \prod_{i:(v_i,t) \in E} \prod_{j=1}^{\min\{x_i, l-1\}} (1 - p_i^{(j)}) \prod_{j=l}^{x_i} (1 - q_i^{(j)}). \quad (3)$$

We denote the probability of $E_{t,l}$ by $\lambda_{t,l}$. Then, the expected number of target nodes activated by the advertiser is $\sum_{t \in T} \sum_l \lambda_{t,l} f_{t,l}(\boldsymbol{x})$, which is monotone and submodular (Soma et al. 2014). When $p_i^{(j)}$ and $q_i^{(j)}$ are nonincreasing with $j$ (i.e., $p_i^{(j)} \ge p_i^{(j+1)}$ and $q_i^{(j)} \ge q_i^{(j+1)}$), the objective $\sum_{t \in T} \sum_l \lambda_{t,l} f_{t,l}(\boldsymbol{x})$ is DR-submodular (Soma et al. 2014).

**Definition 5** (Budget Allocation with a Competitor)**.** *Given a bipartite graph $G = (V, T; E)$, capacities $c_i$ $(i \in [n])$, probabilities $p_i^{(j)}, q_i^{(j)}$ $(i \in [n],\ j \in \{1, 2, \dots, c_i\},\ q_i^{(j)} \le p_i^{(j)})$, and a budget $k$, it is to find a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$ such that*

$$\arg\max\nolimits_{\boldsymbol{x} \le \boldsymbol{c}} \quad \sum\nolimits_{t \in T} \sum\nolimits_l \lambda_{t,l} f_{t,l}(\boldsymbol{x}) \quad s.t. \quad |\boldsymbol{x}| \le k,$$

*where $f_{t,l}(\boldsymbol{x})$ is defined as Eq. (3).*

## The Proposed Approach

In this section, we propose a new approach based on Pareto Optimization (Qian, Yu, and Zhou 2015) for the Multiset Selection problem with size constraints, briefly called POMS. Note that Pareto optimization is a recently emerged framework that uses bi-objective optimization as an intermediate step to solve single-objective optimization problems. It has been successfully applied to the subset selection problem (Friedrich and Neumann 2015; Qian, Yu, and Zhou 2015; Qian et al. 2017c; 2017b) as well as the problem of selecting $k$ pairwise disjoint subsets (Qian et al. 2017a).

POMS reformulates the original problem Eq. (1) as a bi-objective maximization problem

$$\arg\max\nolimits_{\boldsymbol{x} \le \boldsymbol{c}} \quad \big(f_1(\boldsymbol{x}),\ f_2(\boldsymbol{x})\big),$$

$$\text{where } f_1(\boldsymbol{x}) = \begin{cases} -\infty, & |\boldsymbol{x}| \ge 2k \\ f(\boldsymbol{x}), & \text{otherwise} \end{cases}, \quad f_2(\boldsymbol{x}) = -|\boldsymbol{x}|.$$

That is, POMS maximizes the objective function $f$ and minimizes the multiset size $|\boldsymbol{x}|$ simultaneously.

In the bi-objective setting, both the two objective values have to be considered for comparing two solutions $\boldsymbol{x}$ and $\boldsymbol{x}'$. $\boldsymbol{x}$ *weakly dominates* $\boldsymbol{x}'$ (i.e., $\boldsymbol{x}$ is *better* than $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \succeq \boldsymbol{x}'$) if $f_1(\boldsymbol{x}) \ge f_1(\boldsymbol{x}')$ and $f_2(\boldsymbol{x}) \ge f_2(\boldsymbol{x}')$; $\boldsymbol{x}$ *dominates* $\boldsymbol{x}'$ (i.e., $\boldsymbol{x}$ is *strictly better* than $\boldsymbol{x}'$, denoted as $\boldsymbol{x} \succ \boldsymbol{x}'$) if $\boldsymbol{x} \succeq \boldsymbol{x}'$ and either $f_1(\boldsymbol{x}) > f_1(\boldsymbol{x}')$ or $f_2(\boldsymbol{x}) > f_2(\boldsymbol{x}')$. But if neither $\boldsymbol{x}$ is better than $\boldsymbol{x}'$ nor $\boldsymbol{x}'$ is better than $\boldsymbol{x}$, they are *incomparable*.

The procedure of POMS is described in Algorithm 1. It starts from the all-zeros solution $\mathbf{0}$ representing the empty set (line 1), and then iteratively tries to improve the quality of the solutions in the archive $P$ (lines 3-10). In each iteration, a new solution $\boldsymbol{x}'$ is generated by randomly perturbing an archived solution $\boldsymbol{x}$ selected from the current $P$ (lines 4-5); if $\boldsymbol{x}'$ is not dominated by any previously archived solution (line 6), it will be added into $P$, and meanwhile those previously archived solutions weakly dominated by $\boldsymbol{x}'$ will be removed (line 7). Note that the domination-based updating procedure makes $P$ always contain incomparable solutions.

**Definition 6** (Random Perturbation)**.** *Given a solution $\mathbf{0} \le \boldsymbol{x} \le \boldsymbol{c}$, the random perturbation operator generates a new solution $\boldsymbol{x}'$ by independently flipping each position of $\boldsymbol{x}$ with probability $1/n$, where the flipping on one position changes the current value to a different value selected uniformly at random. That is, for all $j \in [n]$,*

$$x'_j = \begin{cases} x_j, & \textit{with probability } 1 - 1/n \\ i, & \textit{otherwise} \end{cases},$$

*where $i$ is uniformly chosen from $\{0, 1, \dots, c_i\} \setminus \{x_j\}$.*

POMS repeats for $T$ iterations. The value of $T$ is a parameter, which could affect the quality of the produced solution. Their relationship will be analyzed in the next section, and we will use the theoretically derived $T$ value in the experiments. After running $T$ iterations, the best solution (i.e., having the largest $f$ value) satisfying the size constraint in $P$ is selected as the final solution (line 11).

Note that in the bi-objective transformation, the goal of setting $f_1$ to $-\infty$ is to exclude overly infeasible solutions, the size of which is at least $2k$. These infeasible solutions having $f_1 = -\infty$ and $f_2 \le -2k$ are dominated by any feasible solution (e.g., the solution $\mathbf{0}$ having $f_1 = 0$ and $f_2 = 0$), and therefore never introduced into the archive $P$.

## Theoretical Analysis

In this section, we prove the general approximation bound of POMS, which are characterized by the introduced submodularity and DR-submodularity ratio, and apply it to the submodular cases.

**Algorithm 1** POMS Algorithm

```
Input: a monotone function $f : \mathbb{Z}_+^V \to \mathbb{R}_+$, a vector $\boldsymbol{c} \in \mathbb{Z}_+^V$ and a budget $k \in \mathbb{Z}_+$
Parameter: the number $T$ of iterations
Output: a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$ with $\boldsymbol{x} \le \boldsymbol{c}$ and $|\boldsymbol{x}| \le k$
Process:
Let $\boldsymbol{x} = \mathbf{0}$ and $P = \{\boldsymbol{x}\}$.
Let $t = 0$.
while $t < T$ do
    Select $\boldsymbol{x}$ from $P$ uniformly at random.
    $\boldsymbol{x'} = RandomPerturbation(\boldsymbol{x})$.
    if $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{x'}$ then
        $P = (P \setminus \{\boldsymbol{z} \in P \mid \boldsymbol{x'} \succeq \boldsymbol{z}\}) \cup \{\boldsymbol{x'}\}$.
    end if
    $t = t + 1$.
end while
return $\arg\max_{\boldsymbol{x} \in P : |\boldsymbol{x}| \le k} f(\boldsymbol{x})$
```

## Submodularity and DR-Submodularity Ratio

In (Soma and Yoshida 2016), it was shown that DR-submodularity is stronger than submodularity. Here, we prove that submodularity is actually equivalent to a weak version of DR-submodularity, as shown in Lemma 2. Note that this equivalent relation has recently been proved over the continuous domain (Bian et al. 2017), but we prove it over the integer lattice independently and differently. Compared with DR-submodularity (i.e., Definition 2), this weak version only requires the diminishing return property (i.e., Eq. (4)) to hold with any $i \in [n]$ with $x_i = y_i$ instead of any $i \in [n]$. Note that they are equivalent for a set function $f : 2^V \to \mathbb{R}$, since for any $\boldsymbol{x} \le \boldsymbol{y} \in 2^V$ (i.e., $\{0,1\}^n$), the position $i$ which can be added by 1 must satisfy that $x_i = y_i = 0$. According to Lemma 2 and Definition 2, we define the submodulairty ratio in Definition 7 as well as the DR-submodularity ratio in Definition 8, which measure to what extent a general function $f$ has submodular and DR-submodular properties, respectively. When $f$ is a set function, they are equivalent, and are consistent with that in (Zhang and Vorobeychik 2016).

**Lemma 1.** *(Soma et al. 2014) Let $f : \mathbb{Z}_+^V \to \mathbb{R}$ be a submodular function. For any $\boldsymbol{x}$ and $\boldsymbol{y}$, we have*

$$f(\boldsymbol{x} \vee \boldsymbol{y}) \le f(\boldsymbol{x}) + \sum_{i \in [n] : y_i - x_i > 0} (f(\boldsymbol{x} + (y_i - x_i)\boldsymbol{\chi_i}) - f(\boldsymbol{x})).$$

**Lemma 2.** *A function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is submodular if and only if for any $\boldsymbol{x} \le \boldsymbol{y}$ and $i \in [n]$ with $x_i = y_i$,*

$$f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x}) \ge f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y}). \quad (4)$$

*Proof.* We first prove the '*only if*' case. That is, we need to prove Eq. (4) by using the submodularity of $f$. For any $\boldsymbol{x} \in \mathbb{Z}_+^V$, $i, j \in [n]$ and $a, b \in \mathbb{Z}_+$, if $i \ne j$, we have

$$\begin{aligned} & f(\boldsymbol{x} + a\boldsymbol{\chi_i} + b\boldsymbol{\chi_j}) - f(\boldsymbol{x} + a\boldsymbol{\chi_i}) \qquad (5) \\ & = f(\boldsymbol{x} + a\boldsymbol{\chi_i} + b\boldsymbol{\chi_j}) - f(\boldsymbol{x}) - (f(\boldsymbol{x} + a\boldsymbol{\chi_i}) - f(\boldsymbol{x})) \\ & \le f(\boldsymbol{x} + b\boldsymbol{\chi_j}) - f(\boldsymbol{x}), \end{aligned}$$

where the inequality can be derived by Lemma 1, since $\boldsymbol{x} + a\boldsymbol{\chi_i} + b\boldsymbol{\chi_j}$ can be equivalently represented by $\boldsymbol{x} \vee ((x_i + a)\boldsymbol{\chi_i} + (x_j + b)\boldsymbol{\chi_j})$. For any $\boldsymbol{x} \le \boldsymbol{y}$, let $I = \{i \in [n] : y_i > x_i\}$. We denote the elements in $I$ by $i_1, i_2, \ldots, i_m$, where $|I| = m$. For any $i \in [n]$ with $x_i = y_i$, obviously $i \notin I$, i.e., $i$ is not equal to any $i_j \in I$. Then, we have

$$\begin{aligned} & f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x}) \\ & \ge f(\boldsymbol{x} + (y_{i_1} - x_{i_1})\boldsymbol{\chi_{i_1}} + \boldsymbol{\chi_i}) - f(\boldsymbol{x} + (y_{i_1} - x_{i_1})\boldsymbol{\chi_{i_1}}) \\ & \ge \cdots \\ & \ge f\Big(\boldsymbol{x} + \sum_{j=1}^{m} (y_{i_j} - x_{i_j})\boldsymbol{\chi_{i_j}} + \boldsymbol{\chi_i}\Big) - f\Big(\boldsymbol{x} + \sum_{j=1}^{m} (y_{i_j} - x_{i_j})\boldsymbol{\chi_{i_j}}\Big) \\ & = f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y}), \end{aligned}$$

where the inequalities are derived by repeatedly applying Eq. (5). Thus, the '*only if*' case holds.

Then, we prove the '*if*' case. According to Definition 1, we only need to prove that for any $\boldsymbol{x}$ and $\boldsymbol{y}$,

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \ge f(\boldsymbol{x} \wedge \boldsymbol{y}) + f(\boldsymbol{x} \vee \boldsymbol{y}).$$

We first extend Eq. (4). For any $\boldsymbol{x} \le \boldsymbol{y}$, $i \in [n]$ with $x_i = y_i$, and $a \in \mathbb{Z}_+$, we have

$$\begin{aligned} & f(\boldsymbol{x} + a\boldsymbol{\chi_i}) - f(\boldsymbol{x}) \qquad (6) \\ & = \sum\nolimits_{j=1}^{a} f(\boldsymbol{x} + j\boldsymbol{\chi_i}) - f(\boldsymbol{x} + (j-1)\boldsymbol{\chi_i}) \\ & \ge \sum\nolimits_{j=1}^{a} f(\boldsymbol{y} + j\boldsymbol{\chi_i}) - f(\boldsymbol{y} + (j-1)\boldsymbol{\chi_i}) \\ & = f(\boldsymbol{y} + a\boldsymbol{\chi_i}) - f(\boldsymbol{y}), \end{aligned}$$

where the inequality can be derived by Eq. (4), since $\boldsymbol{x} + (j-1)\boldsymbol{\chi_i} \le \boldsymbol{y} + (j-1)\boldsymbol{\chi_i}$ and $x_i + j - 1 = y_i + j - 1$. For any $\boldsymbol{x}$ and $\boldsymbol{y}$, let $I = \{i \in [n] : x_i > y_i\}$. We denote the elements in $I$ by $i_1, i_2, \ldots, i_m$. Then, we have

$$\begin{aligned} f(\boldsymbol{x}) - f(\boldsymbol{x} \wedge \boldsymbol{y}) = & \sum_{l=1}^{m} \Big( f\Big((\boldsymbol{x} \wedge \boldsymbol{y}) + \sum\nolimits_{j=1}^{l} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}}\Big) \\ & \qquad - f\Big((\boldsymbol{x} \wedge \boldsymbol{y}) + \sum\nolimits_{j=1}^{l-1} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}}\Big)\Big) \\ \ge & \sum_{l=1}^{m} \Big( f\Big(\boldsymbol{y} + \sum\nolimits_{j=1}^{l} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}}\Big) \\ & \qquad - f\Big(\boldsymbol{y} + \sum\nolimits_{j=1}^{l-1} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}}\Big)\Big) \\ = & f(\boldsymbol{x} \vee \boldsymbol{y}) - f(\boldsymbol{y}). \end{aligned}$$

Note that the inequality is derived by Eq. (6). This is because $(\boldsymbol{x} \wedge \boldsymbol{y}) + \sum_{j=1}^{l-1} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}} \le \boldsymbol{y} + \sum_{j=1}^{l-1} (x_{i_j} - y_{i_j})\boldsymbol{\chi_{i_j}}$; for any $1 \le l \le m$, $x_{i_l} > y_{i_l}$, then the $i_l$-th entry of $\boldsymbol{x} \wedge \boldsymbol{y}$ must be equal to $y_{i_l}$. Thus, the '*if*' case holds. □

**Definition 7** (Submodularity Ratio)**.** *The submodularity ratio of a function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is defined as*

$$\alpha_f = \min_{\boldsymbol{x} \le \boldsymbol{y}, i \in [n] : x_i = y_i} \frac{f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x})}{f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y})}.$$

**Definition 8** (DR-Submodularity Ratio)**.** *The DR-submodularity ratio of a function $f : \mathbb{Z}_+^V \to \mathbb{R}$ is defined as*

$$\beta_f = \min_{\boldsymbol{x} \le \boldsymbol{y}, i \in [n]} \frac{f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x})}{f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y})}.$$

It is easy to see that $\beta_f \le \alpha_f$. Note that $(f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x}))/(f(\boldsymbol{y} + \boldsymbol{\chi_i}) - f(\boldsymbol{y}))$ reaches 1 by letting $\boldsymbol{x} = \boldsymbol{y}$, so $\beta_f \le \alpha_f \le 1$. For a monotone function $f$, we make the following observations:

**Remark 1.** *For a monotone function $f : \mathbb{Z}_+^V \to \mathbb{R}_+$, it holds that 1) $0 \le \beta_f \le \alpha_f \le 1$; 2) $f$ is submodular iff $\alpha_f = 1$; 3) $f$ is DR-submodular iff $\beta_f = \alpha_f = 1$.*

## Approximation Guarantee

We prove the general approximation bound of POMS in Theorem 1, where $\mathbb{E}[T]$ denotes the expected number of iterations and $OPT$ denotes the optimal function value. Let $c_{\max} = \max\{c_i \mid i \in [n]\}$. The idea is to prove two approximation guarantees of $(1 - e^{-\beta_f})$ and $(\alpha_f/2)(1 - e^{-\alpha_f})$, respectively, the larger value of which leads to the desired approximation bound. To prove the $(1 - e^{-\beta_f})$-approximation guarantee, we need Lemma 3, that for any multiset, there always exists one item, the inclusion of which can bring an improvement on $f$ proportional to the current distance to the optimum. To prove the $(\alpha_f/2)(1-e^{-\alpha_f})$-approximation guarantee, we need a similar lemma, i.e., Lemma 4, which is about the average gain by adding multiple copies of one item instead of the gain by adding one item.

**Lemma 3.** *Let $f : \mathbb{Z}_+^V \to \mathbb{R}_+$ be a monotone function. For any $\boldsymbol{x} \in \mathbb{Z}_+^V$, there exists $i \in [n]$ such that*

$$f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x}) \ge \frac{\beta_f}{k}(OPT - f(\boldsymbol{x})).$$

*Proof.* For any $\boldsymbol{x} \le \boldsymbol{y}$, $i \in [n]$ and $a \in \mathbb{Z}_+$, we have

$$\begin{aligned} & f(\boldsymbol{y} + a\boldsymbol{\chi_i}) - f(\boldsymbol{y}) \qquad (7) \\ & = \sum\nolimits_{j=1}^{a} f(\boldsymbol{y} + j\boldsymbol{\chi_i}) - f(\boldsymbol{y} + (j-1)\boldsymbol{\chi_i}) \\ & \le \frac{a}{\beta_f}(f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x})), \end{aligned}$$

where the inequality is by the definition of the DR-submodularity ratio (i.e., Definition 8). Let $\boldsymbol{x}^*$ be an optimal solution, i.e., $f(\boldsymbol{x}^*) = OPT$. For any $\boldsymbol{x}$, let $I = \{i \in [n] : x_i^* > x_i\}$. We denote the elements in $I$ by $i_1, i_2, \ldots, i_m$. Let $i^* = \arg\max_{i \in [n]} f(\boldsymbol{x} + \boldsymbol{\chi_i}) - f(\boldsymbol{x})$. Then, we have

$$\begin{aligned} & f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \le f(\boldsymbol{x}^* \vee \boldsymbol{x}) - f(\boldsymbol{x}) \qquad (8) \\ & = \sum_{l=1}^{m} \Big( f\Big(\boldsymbol{x} + \sum\nolimits_{j=1}^{l} (x_{i_j}^* - x_{i_j})\boldsymbol{\chi_{i_j}}\Big) \\ & \qquad - f\Big(\boldsymbol{x} + \sum\nolimits_{j=1}^{l-1} (x_{i_j}^* - x_{i_j})\boldsymbol{\chi_{i_j}}\Big)\Big) \\ & \le \sum_{l=1}^{m} \frac{x_{i_l}^* - x_{i_l}}{\beta_f}(f(\boldsymbol{x} + \boldsymbol{\chi_{i_l}}) - f(\boldsymbol{x})) \\ & \le \frac{k}{\beta_f}(f(\boldsymbol{x} + \boldsymbol{\chi_{i^*}}) - f(\boldsymbol{x})), \end{aligned}$$

where the first inequality is by the monotonicity of $f$, the second inequality is by Eq. (7), and the last is by $\sum_{l=1}^{m}(x_{i_l}^* - x_{i_l}) \le \sum_{l=1}^{m} x_{i_l}^* \le |\boldsymbol{x}^*| \le k$. Thus, the lemma holds. □

**Lemma 4.** *Let $f : \mathbb{Z}_+^V \to \mathbb{R}_+$ be a monotone function. For any $\boldsymbol{x} \in \mathbb{Z}_+^V$, there exists $i \in [n]$ and $j \le k - x_i$ such that*

$$\frac{f(\boldsymbol{x} + j\boldsymbol{\chi_i}) - f(\boldsymbol{x})}{j} \ge \frac{\alpha_f}{k} \cdot (OPT - f(\boldsymbol{x})).$$

*Proof.* For any $\boldsymbol{x} \le \boldsymbol{y}$, $i \in [n]$ with $x_i = y_i$, and $a \in \mathbb{Z}_+$,

$$f(\boldsymbol{x} + a\boldsymbol{\chi_i}) - f(\boldsymbol{x}) \ge \alpha_f(f(\boldsymbol{y} + a\boldsymbol{\chi_i}) - f(\boldsymbol{y})). \qquad (9)$$

The analysis is as same as Eq. (6), except that the definition of the submodularity ratio (i.e., Definition 7) is applied to derive the inequality. Then, by applying Eq. (9) to the formula at the right of '$=$' in Eq. (8), we get

$$f(\boldsymbol{x}^*) - f(\boldsymbol{x}) \le \sum_{l=1}^{m} \frac{1}{\alpha_f}(f(\boldsymbol{x} + (x_{i_l}^* - x_{i_l})\boldsymbol{\chi_{i_l}}) - f(\boldsymbol{x})).$$

The conditions of Eq. (9) are easily verified, since $\boldsymbol{x} \le \boldsymbol{x} + \sum_{j=1}^{l-1}(x_{i_j}^* - x_{i_j})\boldsymbol{\chi_{i_j}}$ and $x_{i_l}$ must be equal to the $i_l$-th entry of $\boldsymbol{x} + \sum_{j=1}^{l-1}(x_{i_j}^* - x_{i_j})\boldsymbol{\chi_{i_j}}$. Let $(i^*, j^*) = \arg\max_{i \in [n], j \le k - x_i} \frac{f(\boldsymbol{x} + j\boldsymbol{\chi_i}) - f(\boldsymbol{x})}{j}$. Note that $x_{i_l}^* - x_{i_l} \le |\boldsymbol{x}^*| - x_{i_l} \le k - x_{i_l}$. Thus, we have

$$\begin{aligned} f(\boldsymbol{x}^*) - f(\boldsymbol{x}) & \le \sum_{l=1}^{m} \frac{x_{i_l}^* - x_{i_l}}{j^* \alpha_f}(f(\boldsymbol{x} + j^*\boldsymbol{\chi_{i^*}}) - f(\boldsymbol{x})) \\ & \le \frac{k}{j^* \alpha_f}(f(\boldsymbol{x} + j^*\boldsymbol{\chi_{i^*}}) - f(\boldsymbol{x})), \end{aligned}$$

which implies that the lemma holds. □

**Theorem 1.** *For the problem in Definition 3, POMS with $\mathbb{E}[T] \le 2ec_{\max}k^2n$ finds a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$ with $\boldsymbol{x} \le \boldsymbol{c}$, $|\boldsymbol{x}| \le k$ and $f(\boldsymbol{x}) \ge \max\{1 - e^{-\beta_f}, (\alpha_f/2)(1 - e^{-\alpha_f})\} \cdot OPT$.*

*Proof.* **[Part I]** We first prove that POMS with $\mathbb{E}[T] \le 2ec_{\max}k^2n$ can obtain the $(1 - e^{-\beta_f})$-approximation bound. Let $J_{\max}$ denote the maximum value of $j \in \{0, \ldots, k\}$ such that in the archive set $P$, there exists a solution $\boldsymbol{x}$ with $|\boldsymbol{x}| \le j$ and $f(\boldsymbol{x}) \ge (1 - (1 - \frac{\beta_f}{k})^j) \cdot OPT$. That is,

$$\begin{aligned} J_{\max} = \max\{j \in \{0, \ldots, k\} \mid \exists \boldsymbol{x} \in P, |\boldsymbol{x}| \le j \\ \wedge f(\boldsymbol{x}) \ge (1 - (1 - \beta_f/k)^j) \cdot OPT\}. \end{aligned}$$

We then only need to analyze the expected number of iterations until $J_{\max} = k$. This is because $J_{\max} = k$ implies that there exists one solution $\boldsymbol{x}$ in $P$ satisfying that $|\boldsymbol{x}| \le k$ and $f(\boldsymbol{x}) \ge (1 - (1 - \frac{\beta_f}{k})^k) \cdot OPT \ge (1 - e^{-\beta_f}) \cdot OPT$.

The initial value of $J_{\max}$ is 0, since POMS starts from $\boldsymbol{0}$. Assume that currently $J_{\max} = j < k$. Let $\boldsymbol{x}$ be a corresponding solution with the value $j$, i.e., $|\boldsymbol{x}| \le j$ and

$$f(\boldsymbol{x}) \ge (1 - (1 - \beta_f/k)^j) \cdot OPT. \qquad (10)$$

It is easy to see that $J_{\max}$ cannot decrease because deleting $\boldsymbol{x}$ from $P$ (line 7 of Algorithm 1) implies that $\boldsymbol{x}$ is weakly dominated by the newly generated solution $\boldsymbol{x'}$, which must satisfy $|\boldsymbol{x'}| \leq |\boldsymbol{x}|$ and $f(\boldsymbol{x'}) \geq f(\boldsymbol{x})$. By Lemma 3, we know that adding a specific item into $\boldsymbol{x}$ can generate a new solution $\boldsymbol{x'} = \boldsymbol{x} + \boldsymbol{\chi_i}$, which satisfies that $f(\boldsymbol{x'}) - f(\boldsymbol{x}) \geq \frac{\beta_f}{k}(OPT - f(\boldsymbol{x}))$. Then, by using Eq. (10), we get

$$f(\boldsymbol{x'}) \geq (1 - (1 - \beta_f/k)^{j+1}) \cdot OPT.$$

Since $|\boldsymbol{x'}| = |\boldsymbol{x} + \boldsymbol{\chi_i}| = |\boldsymbol{x}| + 1 \leq j+1$, $\boldsymbol{x'}$ will be included into $P$; otherwise, $\boldsymbol{x'}$ must be dominated by one solution in $P$ (line 6 of Algorithm 1), and this implies that $J_{\max}$ has already been larger than $j$, which contradicts with the assumption $J_{\max} = j$. After including $\boldsymbol{x'}$, $J_{\max} \geq j+1$. Let $P_{\max}$ denote the largest size of $P$ during the run of POMS. Thus, $J_{\max}$ can increase by at least 1 in one iteration with probability at least $\frac{1}{P_{\max}} \cdot \frac{1}{nc_i}(1-\frac{1}{n})^{n-1} \geq \frac{1}{enc_i P_{\max}}$, where $\frac{1}{P_{\max}}$ is a lower bound on the probability of selecting $\boldsymbol{x}$ in line 4 of Algorithm 1 due to uniform selection, and $\frac{1}{nc_i}(1-\frac{1}{n})^{n-1}$ is the probability of changing $x_i$ to $x_i + 1$ while keeping other positions unchanged by the random perturbation operator (as shown in Definition 6). Then, it needs at most $enc_i P_{\max}$ iterations in expectation to increase $J_{\max}$. After $k \cdot enc_i P_{\max}$ expected number of iterations, $J_{\max}$ must have reached $k$.

By the procedure of POMS, we know that the solutions maintained in $P$ must be incomparable. Thus, each value of one objective can correspond to at most one solution in $P$. Because the solutions with $|\boldsymbol{x}| \geq 2k$ are excluded from $P$, $|\boldsymbol{x}|$ must belong to $\{0, 1, \ldots, 2k-1\}$, which implies that $P_{\max} \leq 2k$. Furthermore, $c_i \leq c_{\max}$. Hence, the expected number of iterations $\mathbb{E}[T]$ for obtaining the $(1 - e^{-\beta_f})$-approximation guarantee is at most $2ec_{\max}k^2n$.

**[Part II]** We then prove that POMS with $\mathbb{E}[T] \leq 2ec_{\max}k^2n$ can obtain the $(\alpha_f/2)(1 - e^{-\alpha_f})$-approximation bound. We also analyze a quantity $J_{\max}$, which is defined as

$$J_{\max} = \max\{j \in \{0, \ldots, k\} \mid \exists \boldsymbol{x} \in P, |\boldsymbol{x}| \leq j$$
$$\wedge f(\boldsymbol{x}) \geq \big(1 - \big(1 - \alpha_f \tfrac{j}{km}\big)^m\big) \cdot OPT \text{ for some } m \in \mathbb{Z}_+\}.$$

The initial value of $J_{\max}$ is 0. Assume that currently $J_{\max} = j \leq k$. Let $\boldsymbol{x}$ be a corresponding solution with the value $j$, i.e., $|\boldsymbol{x}| \leq j$ and $f(\boldsymbol{x}) \geq (1 - (1 - \alpha_f \frac{j}{km})^m) \cdot OPT$ for some $m$. As the analysis in Part I, $J_{\max}$ cannot decrease. By Lemma 4, we know that for some $i \in [n]$ and $l \leq k - x_i$, increasing $x_i$ by $l$ can generate a new solution $\boldsymbol{x'} = \boldsymbol{x} + l\boldsymbol{\chi_i}$ satisfying that $f(\boldsymbol{x'}) - f(\boldsymbol{x}) \geq \alpha_f \frac{l}{k}(OPT - f(\boldsymbol{x}))$. By applying the lower bound on $f(\boldsymbol{x})$ to this inequality, we get

$$\begin{aligned} f(\boldsymbol{x'}) &\geq \big(1 - \big(1 - \alpha_f \tfrac{j}{km}\big)^m \big(1 - \alpha_f \tfrac{l}{k}\big)\big) \cdot OPT \\ &\geq \Big(1 - \big(1 - \alpha_f \tfrac{j+l}{k(m+1)}\big)^{m+1}\Big) \cdot OPT, \end{aligned}$$

where the second inequality is by applying the AM-GM inequality. Note that $|\boldsymbol{x'}| = |\boldsymbol{x}| + l \leq j + l$. As the analysis in Part I, $\boldsymbol{x'}$ will be included into $P$, which makes $J_{\max} \geq j+l$. Thus, $J_{\max}$ can increase by at least $l$ in one iteration with probability at least $\frac{1}{P_{\max}} \cdot \frac{1}{nc_i}(1 - \frac{1}{n})^{n-1} \geq \frac{1}{2ec_{max}kn}$. That is, it needs at most $2ec_{max}kn$ expected number of iterations to increase $J_{\max}$ by at least $l \geq 1$. After at most $k \cdot 2ec_{max}kn$ expected number of iterations, $J_{\max}$ cannot increase, i.e., $J_{\max} + l > k$. This implies that there exists one solution $\boldsymbol{x}$ in $P$ satisfying that $|\boldsymbol{x}| \leq J_{\max} \leq k$ and for some $i \in [n]$ and $k - J_{\max} < l \leq k - x_i$,

$$\begin{aligned} f(\boldsymbol{x} + l\boldsymbol{\chi_i}) &\geq \Big(1 - \big(1 - \alpha_f \tfrac{J_{\max}+l}{k(m+1)}\big)^{m+1}\Big) \cdot OPT \\ &> \Big(1 - \big(1 - \tfrac{k\alpha_f}{k(m+1)}\big)^{m+1}\Big) \cdot OPT \geq \big(1 - \tfrac{1}{e^{\alpha_f}}\big) \cdot OPT. \end{aligned}$$

Let $\boldsymbol{y} = \arg\max_{i \in [n], j \leq k} f(j\boldsymbol{\chi_i})$. We then have

$$\begin{aligned} f(\boldsymbol{x} + l\boldsymbol{\chi_i}) &= f(\boldsymbol{x}) + (f(\boldsymbol{x} + l\boldsymbol{\chi_i}) - f(\boldsymbol{x})) \\ &\leq f(\boldsymbol{x}) + (f(x_i\boldsymbol{\chi_i} + l\boldsymbol{\chi_i}) - f(x_i\boldsymbol{\chi_i}))/\alpha_f \\ &\leq f(\boldsymbol{x}) + f(\boldsymbol{y})/\alpha_f \leq (f(\boldsymbol{x}) + f(\boldsymbol{y}))/\alpha_f, \end{aligned}$$

where the first inequality is by Eq. (9), the second inequality is by $f(x_i\boldsymbol{\chi_i}) \geq 0$ and $x_i + l \leq k$, and the last is by $\alpha_f \in [0,1]$. Thus, after at most $2ec_{max}k^2n$ iterations in expectation, $P$ will contain a solution $\boldsymbol{x}$ with $|\boldsymbol{x}| \leq k$ and

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \geq \alpha_f(1 - e^{-\alpha_f}) \cdot OPT.$$

Note that $\mathbf{0}$ will always be in $P$, since it has the smallest size 0 and no other solutions can dominate it. Without loss of generality, we assume that $\boldsymbol{y} = j^*\boldsymbol{\chi_{i^*}}$. Thus, $\boldsymbol{y}$ can be generated in one iteration by selecting $\mathbf{0}$ in line 4 of Algorithm 1 and changing its $i^*$-th entry from 0 to $j^*$ in line 5, whose probability is at least $\frac{1}{P_{\max}} \cdot \frac{1}{nc_{i^*}}(1 - \frac{1}{n})^{n-1} \geq \frac{1}{2ec_{\max}kn}$. That is, $\boldsymbol{y}$ will be generated in at most $2ec_{\max}kn$ expected iterations. According to the updating procedure of $P$ (lines 6-8), we know that once $\boldsymbol{y}$ is produced, $P$ will always contain a solution $\boldsymbol{z} \succeq \boldsymbol{y}$, i.e., $|\boldsymbol{z}| \leq |\boldsymbol{y}| \leq k$ and $f(\boldsymbol{z}) \geq f(\boldsymbol{y})$.

By line 11 of Algorithm 1, the best solution satisfying the size constraint will be finally returned. Thus, POMS using $\mathbb{E}[T] \leq 2ec_{max}k^2n$ finds a solution with the $f$ value at least

$$\max\{f(\boldsymbol{x}), f(\boldsymbol{y})\} \geq (\alpha_f/2) \cdot (1 - e^{-\alpha_f}) \cdot OPT.$$

**Taking the better** of the two approximation bounds (i.e., $1 - e^{-\beta_f}$ and $\alpha_f(1 - e^{-\alpha_f})/2$) derived in Parts I and II leads to the desired approximation bound. The required number of iterations is $2ec_{max}k^2n$. It is clear that the random perturbation operator makes any produced solution $\boldsymbol{x}$ in the run of POMS satisfy $\mathbf{0} \leq \boldsymbol{x} \leq \boldsymbol{c}$. Thus, the theorem holds. □

## Applications of Approximation Guarantee

We have shown that POMS can achieve an approximation guarantee of $\max\{1 - e^{-\beta_f}, \alpha_f(1 - e^{-\alpha_f})/2\}$. A natural question is then how good this approximation bound can be. We apply it to the submodular cases. According to Remark 1, we make the following observations:

**Remark 2.** *When the objective $f$ is DR-submodular, the approximation bound of POMS is $1 - e^{-1}$; when $f$ is submodular, the bound is $\max\{1 - e^{-\beta_f}, (1 - e^{-1})/2\}$.*

For $f$ being DR-submodular, the best known approximation bound is $1 - e^{-1}$, which was obtained by the greedy algorithm (Soma et al. 2014). For $f$ being submodular, the generalized greedy algorithm obtains an $(1 - e^{-1})/2$-approximation bound (Alon, Gamzu, and Tennenholtz 2012). Note that although this bound can be improved to $1 - e^{-1}$ by partial enumerations, the runtime is impractical (Soma et al. 2014). Thus, the approximation bound of POMS is at least as good as that of the greedy-style methods.

To further show that the derived approximation guarantee of POMS is applicable to real-world multiset selection tasks, we consider budget allocation and that with a competitor. Their objective functions are known to be submodular (Soma et al. 2014). Thus, $\alpha_f = 1$ and the approximation bound is $\max\{1 - e^{-\beta_f}, (1 - e^{-1})/2\}$. We then only need to give lower bounds on the DR-submodularity ratio $\beta_f$, as shown in Lemmas 5 and 6. When $p_i^{(j)}$ and $q_i^{(j)}$ are nonincreasing with $j$, it is easy to verify that $\beta_f = 1$ in both tasks, that is, the objective functions are DR-submodular. This is consistent with the previous result in (Alon, Gamzu, and Tennenholtz 2012). Due to space limitation, the proofs of Lemma 5 and 6 are provided in the supplementary material.

**Lemma 5.** *For budget allocation in Definition 4, the DR-submodularity ratio can be lower bounded as*

$$\beta_f \geq \min_{i \in [n], 1 \leq j \leq r \leq c_i} p_i^{(j)} / p_i^{(r)}.$$

**Lemma 6.** *For budget allocation with a competitor in Definition 5, the DR-submodularity ratio can be lower bounded as*

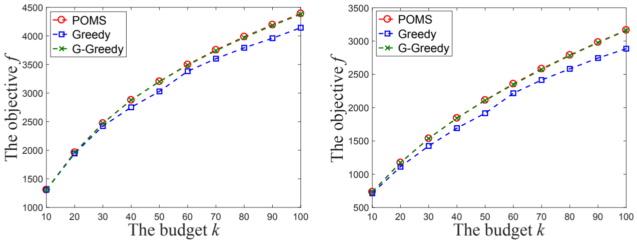
$$\beta_f \geq \min_{i \in [n], 1 \leq j \leq r \leq c_i} \min\{p_i^{(j)} / p_i^{(r)}, q_i^{(j)} / q_i^{(r)}\}.$$

## Experiments

In this section, we empirically compare POMS with the greedy algorithm (Soma et al. 2014) and the generalized greedy algorithm (Alon, Gamzu, and Tennenholtz 2012) on the applications of budget allocation and generalized influence maximization. These two greedy methods are briefly called Greedy and G-Greedy, respectively. The number $T$ of iterations of POMS is set to $2ec_{\max}k^2n$ as suggested by Theorem 1. As POMS is a randomized algorithm, we repeat the run 10 times independently and report the average results.

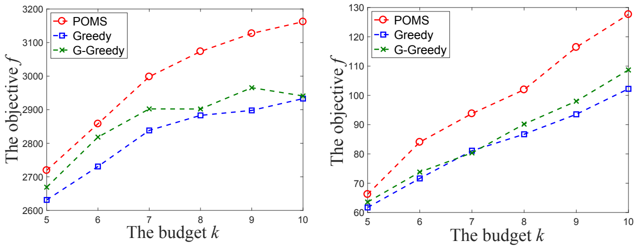
**Budget Allocation.** We use one real-world data set Yahoo! Search Marketing Advertiser Bidding Data[1], which is a bipartite graph representing "which customers are interested in which keywords". It contains $n$ = 1,000 source nodes (i.e., keywords), 10,475 target nodes (i.e., customers) and 52,567 edges. For each influence probability $p_i^{(j)}$ of each source node $v_i$, we use a randomly generated value between 0 and 0.5. The capacities $c_i$ are set to 5 (thus $c_{\max} = 5$). The budget $k$ is set from 10 to 100. For the case with a competitor, the probability $q_i^{(j)}$ is set to $0.2 \cdot p_i^{(j)}$, and the budget of the competitor is set to 100, which is allocated to the top 100 highest degree source nodes in advance.

**Generalized Influence Maximization.** Let a directed graph $G = (V, E)$ represent a social network, where each node is a user and each edge $(v_i, v_j) \in E$ has a probability $p_{i,j}$ representing the influence strength from user $v_i$ to $v_j$. Given a budget $k$, influence maximization is to find a subset $\boldsymbol{x} \in 2^V$ of size $k$ such that the expected number of nodes activated by propagating from $\boldsymbol{x}$ is maximized (Kempe, Kleinberg, and Tardos 2003). As it is natural to assign different budgets to different users in practice, we introduce the generalized influence maximization problem, which allows to select one user multiple times. That is, we are to select a multiset $\boldsymbol{x} \in \mathbb{Z}_+^V$, and $x_i$ is the budget allocated to the user $v_i$. Each user $v_i$ with the budget $x_i > 0$ will make $x_i$ independent trials of diffusion. The set of nodes to propagate in the $l$-th trial is $X_l = \{v_i \mid x_i \geq l\}$. Each $X_l$ will propagate independently using probabilities $p_{i,j}$, and the set of nodes that get activated is denoted as $A(X_l)$, which is a random variable. Generalized influence maximization is to maximize the expected total number of nodes that get activated in at least one propagation process. The Independent Cascade propagation model (Goldenberg, Libai, and Muller 2001) is used. The objective function $f(\boldsymbol{x}) = \mathbb{E}\big[|\bigcup_{l \geq 1} A(X_l)|\big]$ is monotone submodular, the proof of which is shown in the supplementary material due to space limitation.



(a) without a competitor (b) with a competitor

Figure 1: Budget allocation on the *Yahoo!* data set.



(a) *ego-Facebook* (b) *Weibo*

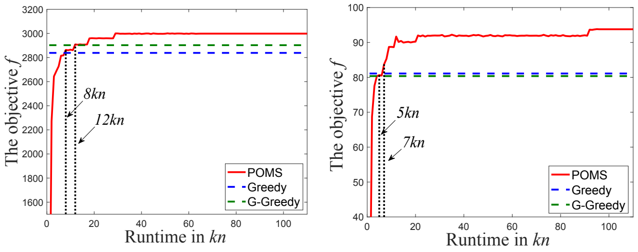
Figure 2: Generalized influence maximization.

We use two real-world data sets: *ego-Facebook*[2] (4,039 nodes, 88,234 edges) and *Weibo* (5,000 nodes, 65,148 edges). *Weibo* is crawled from a Chinese microblogging site Weibo.com like Twitter. On each network, the probability of one edge from $v_i$ to $v_j$ is estimated by $\frac{weight(v_i, v_j)}{indegree(v_j)}$, as widely used in (Chen, Wang, and Yang 2009; Goyal, Lu, and Lakshmanan 2011). The capacities $c_i$ are set to 5. The budget $k$ is set from 5 to 10. To estimate the expected number of active nodes, we simulate the diffusion process 30 times independently and use the average as an estimation. But for the final output solutions of the algorithms, we average over 10,000 times for more accurate estimation.

**Results.** The results are plotted in Figures 1 and 2, which show that POMS is never worse. On generalized influence maximization, POMS performs much better in most cases. This is expected. POMS maintains multiple solutions in the optimization process; in each iteration, it can add different items simultaneously and can also delete some selected

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=a

[2] http://snap.stanford.edu/data/index.html



(a) *ego-Facebook* (b) *Weibo*

Figure 3: Performance v.s. runtime of POMS.

items. These characteristics may make POMS easier than Greedy and G-Greedy to escape from local optima. We also note that on budget allocation, the performances of POMS and G-Greedy are almost same. This may be because G-Greedy has already been nearly optimal, due to the simple network structure (i.e., a bipartite graph) of the budget allocation problem.

Considering the runtime (in the number of objective evaluations), Greedy and G-Greedy take the time in the order of $kn$ and $c_{\max}kn$, respectively; POMS is set to use $2ec_{\max}k^2n$ time according to the theoretical upper bound (i.e., the worst-case time) for POMS being good. By selecting Greedy and G-Greedy as the baseline, we plot the curve of the objective $f$ over the time for POMS on generalized influence maximization with $k=7$, as shown in Figure 3. We can see that the time of POMS to obtain a better performance is much less than the worst-case time $2ec_{\max}k^2n \approx 190kn$. This implies that POMS can be efficient in practice.

## Conclusion

In this paper, we consider the multiset selection problem with size constraints. We relax the submodularity assumption of previous studies, and propose a new algorithm POMS, which can achieve a good approximation guarantee, characterized by the introduced submodularity ratio. The experimental results on two real-world applications show the superior performance of POMS.

## References

Alon, N.; Gamzu, I.; and Tennenholtz, M. 2012. Optimizing budget allocation among channels and influencers. In *WWW'12*, 381–388.

Bian, A. A.; Mirzasoleiman, B.; Buhmann, J.; and Krause, A. 2017. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *AISTATS'17*, 111–120.

Chen, W.; Wang, Y.; and Yang, S. 2009. Efficient influence maximization in social networks. In *KDD'09*, 199–208.

Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM* 45(4):634–652.

Friedrich, T., and Neumann, F. 2015. Maximizing submodular functions under matroid constraints by evolutionary algorithms. *Evolutionary Computation* 23(4):543–558.

Goldenberg, J.; Libai, B.; and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters* 12(3):211–223.

Gottschalk, C., and Peis, B. 2015. Submodular function maximization on the bounded integer lattice. In *Proceedings of the 13th International Workshop on Approximation and Online Algorithms (WAOA'15)*, 133–144.

Goyal, A.; Lu, W.; and Lakshmanan, L. V. 2011. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM'11*, 211–220.

Kapralov, M.; Post, I.; and Vondrák, J. 2013. Online submodular welfare maximization: Greedy is optimal. In *SODA'13*, 1216–1225.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *KDD'03*, 137–146.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research* 9:235–284.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming* 14(1):265–294.

Qian, C.; Shi, J.-C.; Tang, K.; and Zhou, Z.-H. 2017a. Constrained monotone $k$-submodular function maximization using multi-objective evolutionary algorithms with theoretical guarantee. *IEEE Transactions on Evolutionary Computation*.

Qian, C.; Shi, J.-C.; Yu, Y.; Tang, K.; and Zhou, Z.-H. 2017b. Subset selection under noise. In *NIPS'17*.

Qian, C.; Shi, J.-C.; Yu, Y.; and Tang, K. 2017c. On subset selection with general cost constraints. In *IJCAI'17*, 2613–2619.

Qian, C.; Yu, Y.; and Zhou, Z.-H. 2015. Subset selection by Pareto optimization. In *NIPS'15*, 1765–1773.

Soma, T., and Yoshida, Y. 2015. A generalization of submodular cover via the diminishing return property on the integer lattice. In *NIPS'15*, 847–855.

Soma, T., and Yoshida, Y. 2016. Maximizing monotone submodular functions over the integer lattice. In *Proceedings of the 18th International Conference on Integer Programming and Combinatorial Optimization (IPCO'16)*, 325–336.

Soma, T., and Yoshida, Y. 2017. Non-monotone DR-submodular function maximization. In *AAAI'17*, 898–904.

Soma, T.; Kakimura, N.; Inaba, K.; and Kawarabayashi, K.-i. 2014. Optimal budget allocation: Theoretical guarantee and efficient algorithm. In *ICML'14*, 351–359.

Zhang, H., and Vorobeychik, Y. 2016. Submodular optimization with routing constraints. In *AAAI'16*, 819–826.